

intive



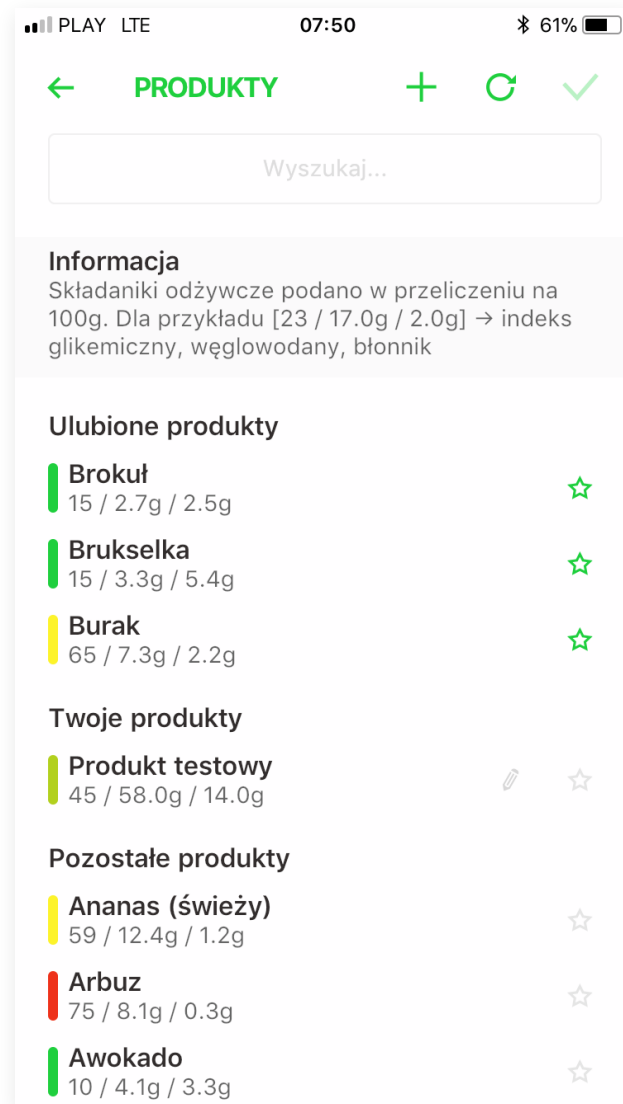
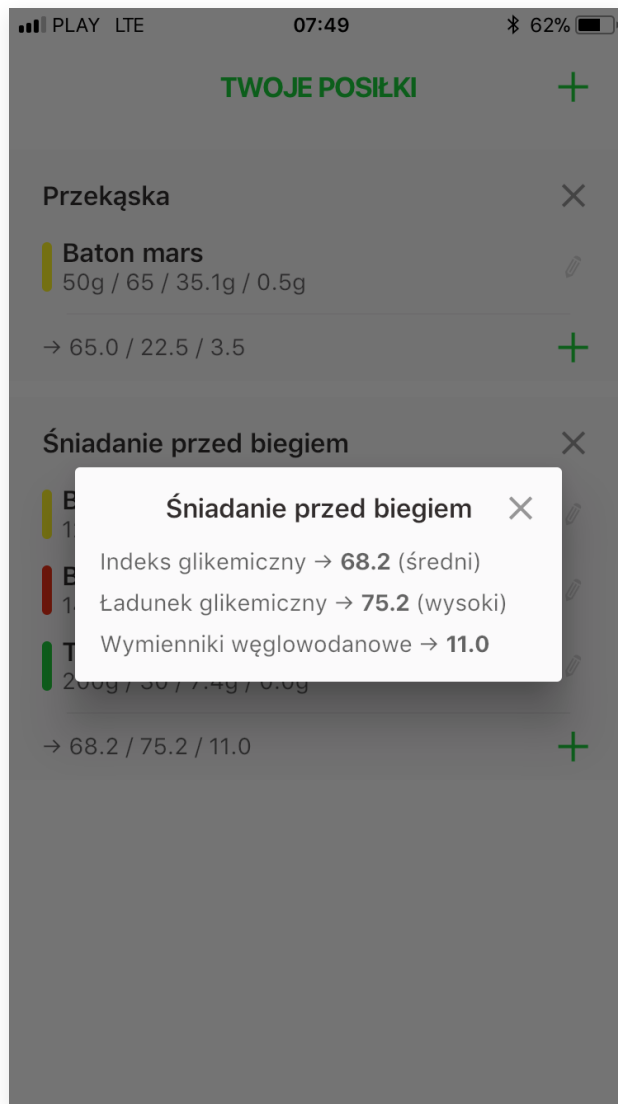
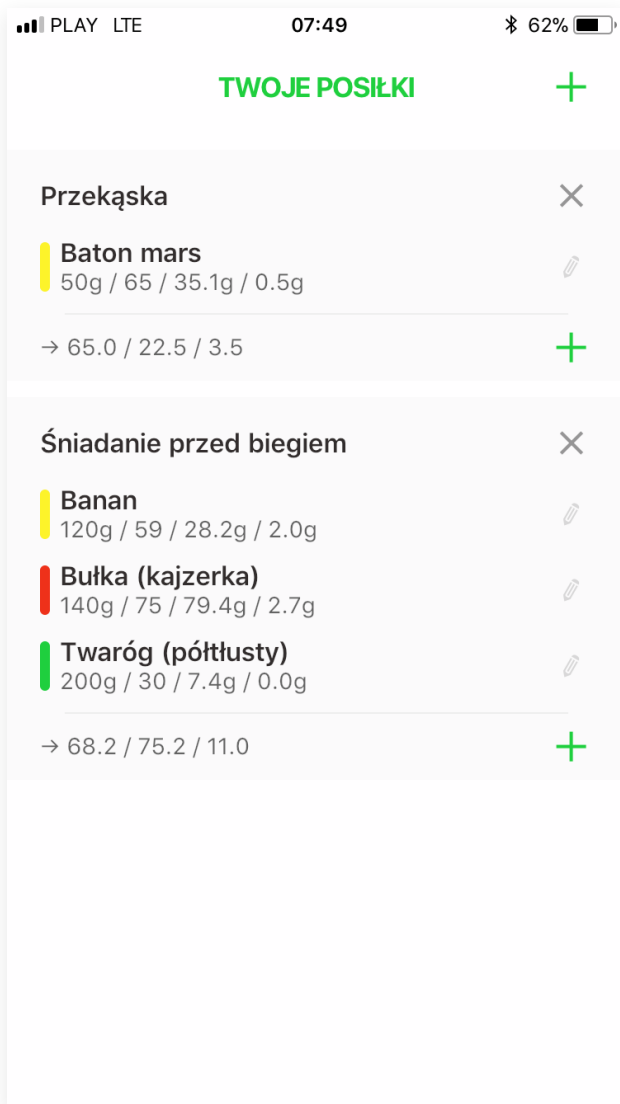
Building MVP with Flutter

TOMASZ SZULC



Project

- _ Android/iOS app - for my girlfriend 🏃
- _ Available for Android 📱
- _ Main features: calculation of glycemic index, glycemic load and carbohydrates exchanges of a meal
- _ To be released on Google Play and App Store after adding more features that integrates the app with her bussiness.



Trelloboard for tracking progress of MVP and ideas 💡

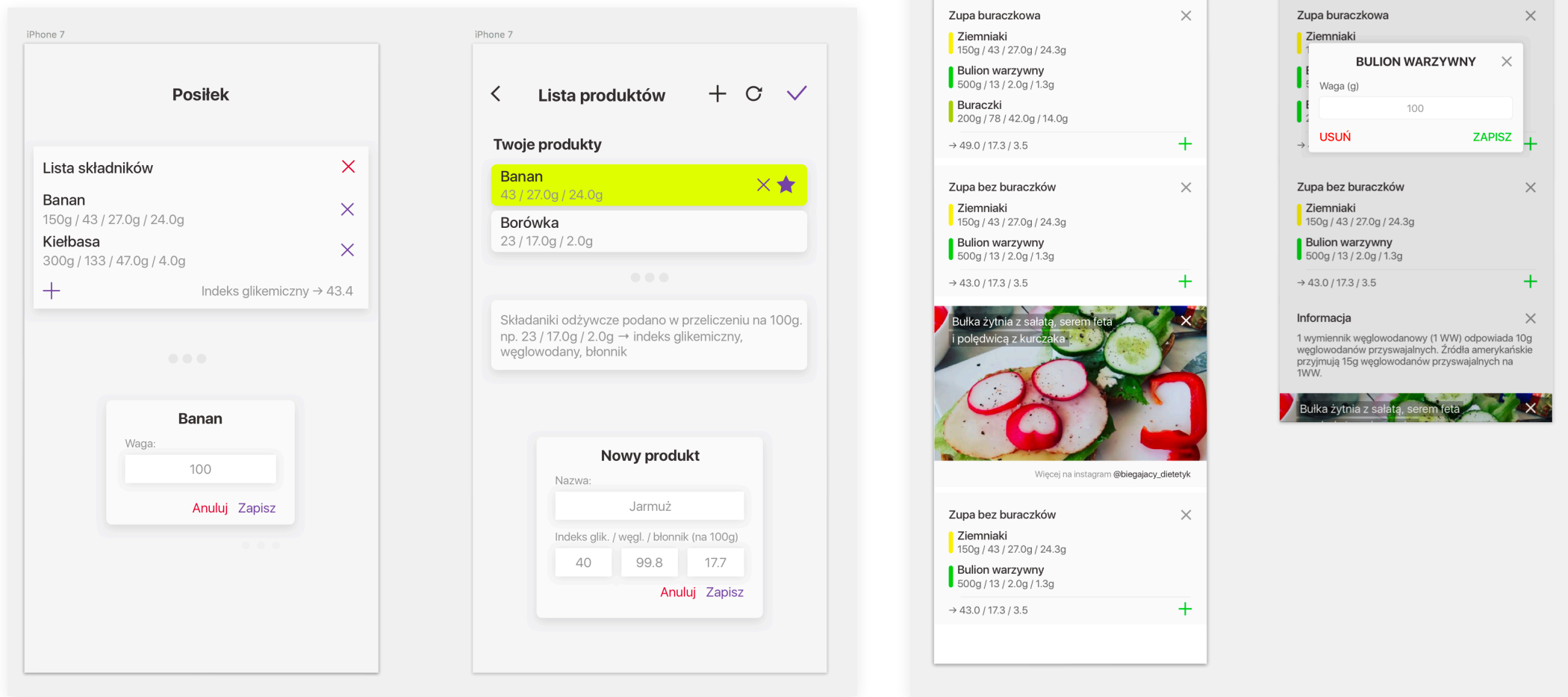
The screenshot shows a Trello board titled "Kalkulator Indeksu Glikemicznego" (Glycemic Index Calculator) in a private workspace. The board is organized into several columns representing different stages of development:

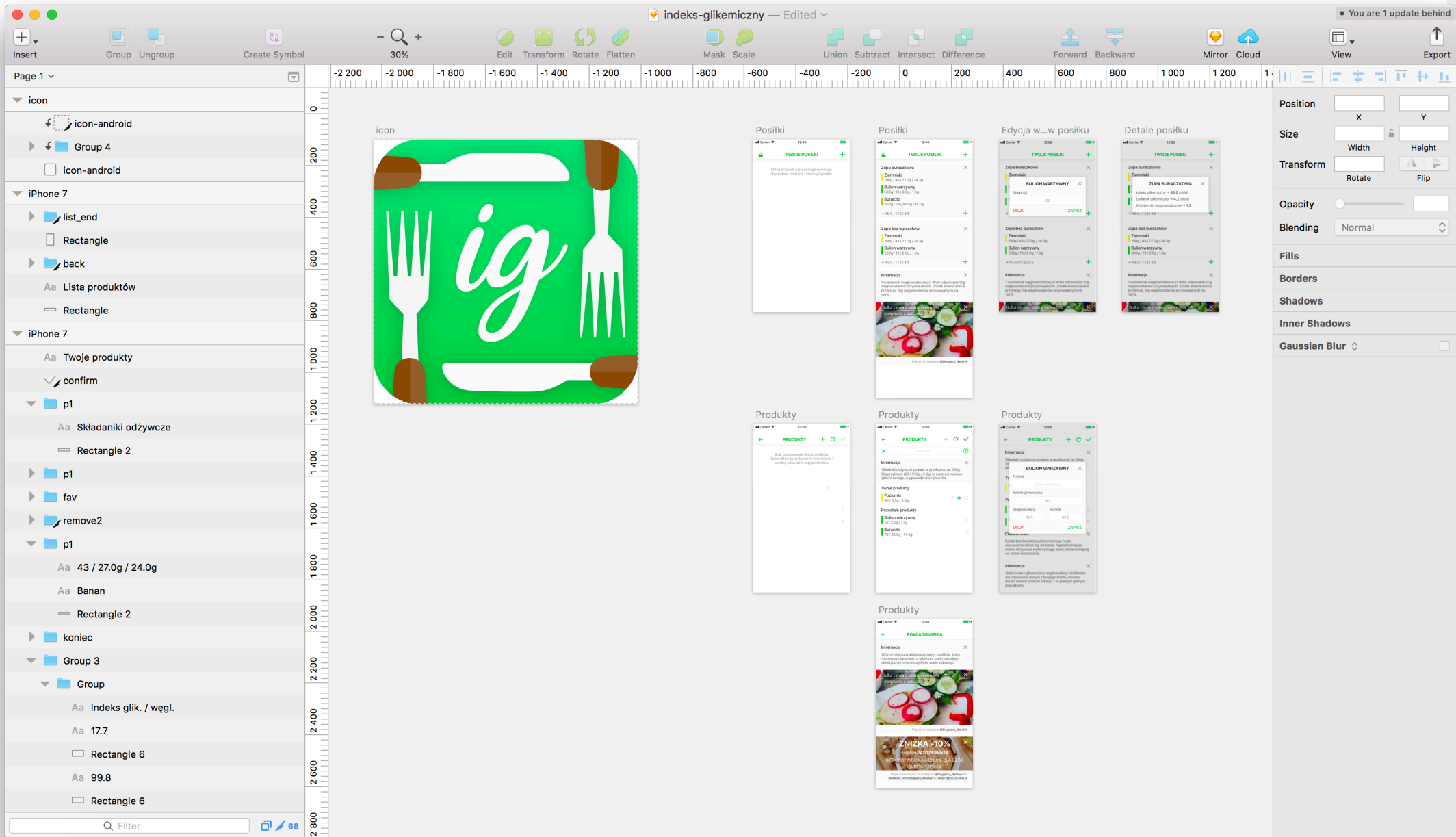
- Skrzynka (Pomysły)** (Ideas):
 - Jako użytkownik chciałbym zobaczyć przeprojektowaną listę produktów z podziałem na sekcje "wszystkie", "twoje", "ulubione", "domyślne"
 - Naprawić sortowanie. Ś znajduje się za Z.
 - Jako użytkownik chcę widzieć delete popup przy usuwaniu posiłku
 - Jako użytkownik chcę przywołać posiłek z historii w celu wyliczenia podobnego posiłku
 - Narysować nową ikonkę dla powiadomień.
 - Dodać odpowiedni font w aplikacji
 - Jako użytkownik chcę widzieć jaki popełniłem błąd przy wypełnianiu formularza dodawania nowego produktu
 - Jako właściciel aplikacji chciałbym wiedzieć jakie produkty dodają użytkownicy by móc powiększyć bazę domyślnych produktów dla przyszłych użytkowników.
- Zablokowane** (Blocked):
 - Jako aplikacja chcę pobierać najnowszą konfigurację aplikacji i przechowywać lokalnie
 - Zaimplementować prezentowanie powiadomień na ekranie posiłków.
 - Jako użytkownik chcę widzieć listę powiadomień
 - Jako Klaudia chcę widzieć swoją reklamę na ekranie głównym, który przekieruje mnie na instagram lub facebook
 - Zmienić nazwę repozytorium i umieścić tam projekt razem z danymi domyślnymi
 - Sprawdzić memory leak gdy przy rozpoczęciu pobierania produktów wychodzimy z widoku i czekamy na timeout
- Brakujące produkty** (Missing products):
 - czekolada gorzka 70% kakao - IG 25
 - czekolada gorzka 85% kakao - IG 20
 - suszone pomidory
- Skrzynka (Priorytety)** (Priority box):
 - Dodawanie nowego produktu lub jego edycja powinny odbywać się na nowym ekranie na Android
- W trakcie** (In progress):
 - + Dodaj kartę
- Ukończone (MVP)** (Completed (MVP)):
 - Stworzyć inną ikonkę dla Android
 - Jak aplikacja działa na innych iPhone?
 - Jak aplikacja działa na urządzeniach z Androidem?
 - Włączanie listy produktów szarpie
 - Wymyślić schemat JSON w jaki sposób aplikacja ma pobierać nowe informacje o powiadomieniach, reklamach itp.
 - Wyłączyć "Favourite" przy rezultatach wyszukiwania
 - Jako użytkownik chcę mieć graficzną reprezentację poziomu indeksu glikemicznego produktu.

Design

- Handcrafted with Sketch :)
- Few iterations and review - even with yourself only, or show it to someone
- Neither iOS nor Android design - custom components for faster development on both platforms - later you can always change the designs.
- The only platform specific component is AppBar.

Design - iterations





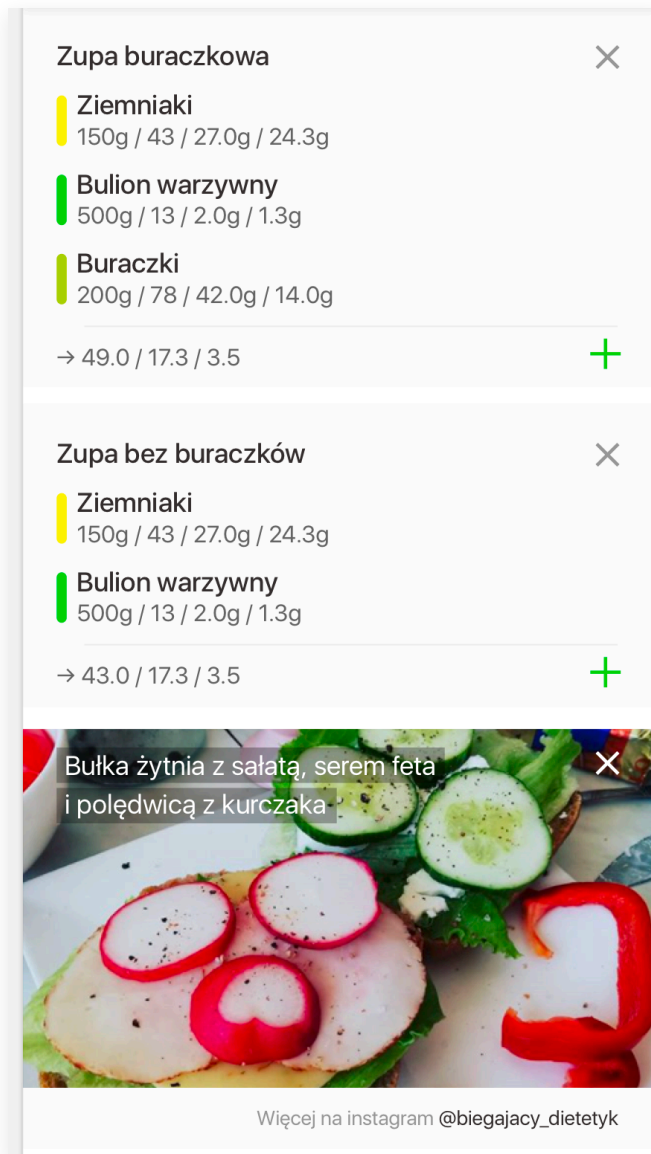
App Development

- _ No plan for MVP at the beginning - Bad 😞 But specified later (Trello) ✌️
- _ No in app storage in the beginning - No Firebase or other modern data storage solutions. App only cached downloaded data in the memory at the beginning.
- _ Download CSV and persist on disk
- _ Download SQLite database and persist on disk
- _ Keep own SQLite database and download other one from server to have default data from the server and be able to add and store other products and meals in app.
- _ Read only server = github :)

What's next?

- _ Github still serves as a server :)
- _ Configuration file on the server downloaded on launch.
- _ Curated list of in-app notifications (messages, new posts on instagram, new articles on the blog, etc.) stored on the github and downloaded from time to time by the app while foregrounded.
- _ Basic analytics
- _ New products in database

```
{
  "id": 1,
  "type": "instagram",
  "created_at": 1540365088,
  "title": "Bułka żytnia z sałatą, serem feta i polędwicą z kurczaka",
  "image": "https://raw.githubusercontent.com/tomkowz/glycemic-index-calculator-data/master/static/1.jpg",
  "external": "https://www.instagram.com/p/Bo3gqlznKpY/?taken-by=biegajacy_dietetyk"
}
```



Flutter

- Great for prototyping! No need to spend a lot of time polishing UI
- You can have a common UI for both platforms
- No need to know Android specific things to create an app if you know iOS. Lots of available packages that supports files storage, preferences and stuff.
- You can later go native with UI and project when your product get a traction and there is future for it. Might lower a cost of development and design at the beginning.

Flutter



- _ Everything is a widget. Components consists of widgets, App consists of components, and app is a widget too :)
- _ Button is a widget, padding is a widget, gesture detector is a widget.
- _ Some time needed to learn about all the widget classes but easy to start and see effects immediately. Complex classess hierarchy.
- _ Many small releases to the framework.
- _ 350 new and 905 closed issues on flutter/flutter since I started working on app.
- _ 41k+ stars.

Flutter

14 hours ago	v0.11.6 ...	d44aa57 zip tar.gz
2 days ago	v0.11.5 ...	97e0310 zip tar.gz
3 days ago	v0.11.4 ...	df57dc9 zip tar.gz
5 days ago	v0.11.3 ...	72bf075 zip tar.gz
5 days ago	v0.11.2 ...	e32cd85 zip tar.gz
6 days ago	v0.11.1 ...	756d593 zip tar.gz
16 days ago	v0.11.0 ...	c319b89 zip tar.gz
20 days ago	v0.10.2 ...	d8cbb80 zip tar.gz
27 days ago	v0.10.1 ...	6a3ff01 zip tar.gz

on 2 Oct	v0.9.6 ...	13684e4 zip tar.gz
on 26 Sep	v0.9.5 ...	020fd59 zip tar.gz
on 25 Sep	v0.9.4 ...	f37c235 zip tar.gz
on 21 Sep	v0.9.3 ...	eed8c7a zip tar.gz
on 19 Sep	v0.9.2 ...	85b4670 zip tar.gz
on 18 Sep	v0.9.1 ...	bf7c270 zip tar.gz
on 18 Sep	v0.9.0 ...	f8c50ea zip tar.gz
on 18 Sep	v0.8.7 ...	ccd070a zip tar.gz
on 17 Sep	v0.8.6 ...	9f359ae zip tar.gz
on 16 Sep	v0.8.5 ...	2bca800 zip tar.gz

Flutter

- _ Dart language; Easy to understand and start with; similar to Swift.
- _ Many libs available for Dart. I had no issue to find libs for my project, but my project is rather very simple case.
- _ My app utilizes following libs: csv, flutter_launcher_icons, http, path_provider, shared_preferences, sqflite, uuid.

Good practices

- _ Start coding new widget as a StatelessWidget and go with StatefulWidget later when needed. Great for prototyping a view to see if it fits and to play with.
- _ Prototype in one file, later you can extract code to separate file and import it. Think locally.

Good practices

Use async/await at the beginning of coding. Speeds up prototyping. Later convert to futures and async code where necessary.

```
117 _initialLoadDataFromDisk() async {
118   final ProductStore store = ProductStore();
119   if (await store.canLoad() == false) {
120     print("Couldn't find product store.");
121     final csv = await ProductDownloader.download();
122     await store.save(csv);
123     print("Did download store from the server.");
124   } else {
125     print("Did find product store on disk.");
126   }
127
128   this._products = await store.load();
129   print("Did load store from disk: ${this._products.length} item(s).");
130
131   // Update UI
132   setState(() => this._isLoading = false);
133 }
134
135 _downloadDataFromServer() async {
136   print("Downloading newest data.");
137   setState(() => this._isLoading = true);
138
139   final csv = await ProductDownloader.download();
140   print(csv);
141   final ProductStore store = ProductStore();
142   await store.save(csv);
143
144   this._products = await store.load();
145   print("Did load store from disk: ${this._products.length} item(s).");
146
147   setState(() => this._isLoading = false);
148 }
149 }
```

Async/await

```
Future<void> deleteProductInMeal(Meal meal, ProductInMeal productInMeal) {
    if (meal.products.length > 1) {
        return this.appContext.appDatabase
            .then((db) async {
                await ProductInMealProvider(db).delete(productInMeal);
                meal.products.remove(productInMeal);
                if (meal.products.isEmpty == true) {
                    return MealProvider(db).delete(meal);
                } else {
                    return null;
                }
            })
            .then((_) => this.loadMeals());
    } else {
        return this.deleteMeal(meal);
    }
}

Future<void> updateMealName(Meal meal, String name) {
    if (meal.name == name) { return null; }

    return this.appContext.appDatabase
        .then((db) => MealProvider(db))
        .then((provider) {
            meal.name = name;
            return provider.update(meal);
        })
        .then((_) => this.loadMeals());
}

Future<void> updateWeightOf(ProductInMeal productInMeal, int weight) {
    if (productInMeal.weight == weight) { return null; }

    return this.appContext.appDatabase
        .then((db) => ProductInMealProvider(db))
        .then((provider) {
            productInMeal.weight = weight;
            return provider.update(productInMeal);
        })
        .then((_) => this.loadMeals());
}

Future<void> addProducts(Meal meal, List<Product> products, int weight) {
    return this.appContext.appDatabase
        .then((db) => MealProvider(db).addProducts(meal, products, weight))
        .then((_) => this.loadMeals());
}
```

```
Future<void> synchronizeDefaultProducts(Database localDatabase) {
    print('Start downloading products from the server.');
```

```
    return NetworkClient().get(this.productsDatabaseUrl)
        .then((response) async {
            print('Finished downloading products from the server.');
```

```
            final bytes = await consolidateHttpClientResponseBytes(response);
            File file = new File((await this.productsDatabaseFilePath).path);
            return file.writeAsBytes(bytes);
        })
        .then((file) => openDatabase(file.path, singleInstance: true, readOnly: true))
        .then((defaultsDatabase) => this._sync(localDatabase, defaultsDatabase));
}

Future<void> _sync(Database localDatabase, Database defaultsDatabase) async {
    print('Start syncing default products.');
```

```
    final allDownloaded = await DefaultProductProvider(db: defaultsDatabase).getAll();
    print('Did download ${allDownloaded.length} products.');
```

```
    final localsProvider = ProductProvider(db: localDatabase);

    return Future.forEach<DefaultProduct>(allDownloaded, (downloaded) async {
        final existing = await localsProvider.getWithUniqueId(downloadId);
        return (existing != null)
            ? this._update(existing, downloaded, localsProvider)
            : this._insert(downloadId, localsProvider);
    }).whenComplete(() {
        defaultsDatabase.close();
        print('Finished syncing.');
```

```
    });
}

Future<void> _update(Product existing, DefaultProduct downloaded, ProductProvider provider) {
    existing.name = downloaded.name;
    existing.glycemicIndex = downloaded.glycemicIndex;
    existing.carbohydrates = downloaded.carbohydrates;
    existing.fiber = downloaded.fiber;
    return provider.update(existing);
}

Future<void> _insert(DefaultProduct downloaded, ProductProvider provider) {
    final product = Product(
        downloaded.uniqueId,
        downloaded.name,
        downloaded.glycemicIndex,
        downloaded.carbohydrates,
        downloaded.fiber,
        true, // is default
        id: downloaded.id,
        isFavourite: false,
    );

    return provider.insert(product);
}
```

Good practices

— Define callbacks as a instance methods to keep the code clean.

```
content = GestureDetector(  
  onTap: () => this._onTap(),  
  onTapDown: (_) => this._onTapDown(),  
  onTapCancel: () => this._onTapCancel(),  
  behavior: HitTestBehavior.opaque,  
  child: content  
);  
  
return content;  
}  
  
_onTap() {  
  this.setState(() => this._isTappedDown = false);  
  this.widget.onTap();  
}  
  
_onTapDown() {  
  this.setState(() => this._isTappedDown = true);  
}  
  
_onTapCancel() {  
  this.setState(() => this._isTappedDown = false);  
}
```

Good practices

DO NOT build widgets like in
Flutter Cookbook on flutter
webpage.

```
123
124 class EditWeightDialog extends StatefulWidget {
125   EditWeightDialog({ Key key }) : super(key: key);
126
127   @override
128   EditWeightDialogState createState() => EditWeightDialogState();
129 }
130
131 class EditWeightDialogState extends State<EditWeightDialog> {
132   final TextEditingController controller = TextEditingController();
133
134   @override
135   Widget build(BuildContext context) {
136     return AlertDialog(
137       contentPadding: const EdgeInsets.all(15.0),
138       content: Row(
139         children: <Widget>[
140           Expanded(
141             child: TextField(
142               controller: controller,
143               keyboardType: TextInputType.number,
144               autofocus: true,
145               decoration: InputDecoration(
146                 labelText: 'Waga [kg]',
147                 hintText: 'np. 150',
148               ), // InputDecoration
149             ), // TextField
150           ), // Expanded
151         ], // <Widget>[]
152       ), // Row
153       actions: <Widget>[
154         FlatButton(
155           child: Text('Anuluj'),
156           onPressed: () => Navigator.pop(context),
157         ), // FlatButton
158         FlatButton(
159           child: Text('Zapisz'),
160           onPressed: () {
161             Navigator.pop(context, controller.text);
162           },
163         ), // FlatButton
164       ], // <Widget>[]
165     ); // AlertDialog
166   }
167 }
```


Good practices

__ Separate code in smaller pieces and add one widget to another.

```
@override
Widget build(BuildContext context) {
  List<Widget> children = [this._buildRibbon(), this._buildDetails(), this._buildButtons()];

  final constraints = BoxConstraints(minHeight: 46.0);

  Widget container = Row(children: children);
  container = Container(constraints: constraints, child: container);
  container = Padding(padding: EdgeInsets.only(left: 20.0, right: 13.0), child: container);
  container = Container(color: this._backgroundColor, child: container);

  return container;
}

Color get _backgroundColor {
  return this._isTappedDown || this.widget._isSelected
    ? AppColors.productCellSelectedBackground
    : AppColors.productCellDeselectedBackground;
}

Widget _buildRibbon() {
  return ColorRibbon(this.widget.glycemicIndex);
}
```

```
Widget _buildDetails() {
  final nameLabel = Text(capitalize(this.widget.productName), style: AppTextStyles.header1);

  final glycemicString = '${this.widget.glycemicIndex.value.toInt()}';
  final carboString = '${this.widget.carbohydrates.toStringAsFixed(1)}g';
  final fiberString = '${this.widget.fiber.toStringAsFixed(1)}g';

  final infoLabel = Text('$glycemicString / $carboString / $fiberString', style: AppTextStyles.text1);

  List<Widget> children = [nameLabel, infoLabel];
  Widget container = Column(
    mainAxisAlignment: MainAxisAlignment.center,
    crossAxisAlignment: CrossAxisAlignment.start,
    children: children,
  );

  container = Padding(padding: EdgeInsets.only(left: 6.0), child: container);

  container = GestureDetector(
    onTapDown: (_) => this._onTapDown(),
    onTapCancel: () => this._onTapCancel(),
    onTapUp: (_) => this._onTapCancel(),
    onTap: () => this._onTap(),
    behavior: HitTestBehavior.opaque,
    child: container
  );

  container = Expanded(child: container);
}
```

Good practices

— „flutter run” starts in debug mode by default. It slows the app a lot! Launch the app with „release” option to see it working full speed.

— When you have multiple devices available in your network you must type identifier of a device to launch the app on it.

— „flutter emulators”, „flutter devices”

— „flutter run -d 3020fcc47e009e817d2b4be2c631fa6992d90b18

— Visual Studio Code with Flutter extension is good IDE.

Programming on Mac using Microsoft IDE to make app for Android ヽ(ゝ)ノ

Issues

Declaring assets catalog didn't work for me. I had to declare every single asset manually.

```
flutter:  
  uses-material-design: false  
  assets:  
    - assets
```

```
flutter:  
  uses-material-design: false  
  assets:  
    - assets/add/add.png  
    - assets/add-2/add-2.png  
    - assets/add-3/add-3.png  
    - assets/back/back.png  
    - assets/bowl/bowl.png  
    - assets/close/close.png  
    - assets/confirm/confirm.png  
    - assets/edit/edit.png  
    - assets/favourite-active/favourite-active.png  
    - assets/favourite-inactive/favourite-inactive.png  
    - assets/refresh/refresh.png  
    - assets/remove/remove.png  
    - assets/remove2/remove2.png  
    - assets/remove-3/remove-3.png  
    - assets/remove-4/remove-4.png  
    - assets/search/search.png
```

Issues

— No Swift-like extensions, so additional code landed in new classes with similar name or global methods.

```
class AppImages {
  static Image get add3 => Image.asset('assets/add-3/add-3.png');
  static Image get back => Image.asset('assets/back/back.png');
  static Image get bowl => Image.asset('assets/bowl/bowl.png');
  static Image get close => Image.asset('assets/close/close.png');
  static Image get confirm => Image.asset('assets/confirm/confirm.png');
  static Image get edit => Image.asset('assets/edit/edit.png');
  static Image get favouriteActive => Image.asset('assets/favourite-active/favourite-active.png');
  static Image get favouriteInactive => Image.asset('assets/favourite-inactive/favourite-inactive.png');
  static Image get refresh => Image.asset('assets/refresh/refresh.png');
  static Image get remove3 => Image.asset('assets/remove-3/remove-3.png');
  static Image get remove4 => Image.asset('assets/remove-4/remove-4.png');
  static Image get search => Image.asset('assets/search/search.png');

  // 1.0
  static Image get add => Image.asset('assets/add/add.png');
  static Image get add2 => Image.asset('assets/add-2/add-2.png');
  static Image get remove => Image.asset('assets/remove/remove.png');
  static Image get remove2 => Image.asset('assets/remove2/remove2.png');
}

class AppColors {
  static Color get navBarBackground => _colorFFFFFF;
  static Color get navBarTitle => _color1FD03E;
  static Color get navBarShadow => _colorDCDCDC_60;

  static Color get screenBackground => _colorFFFFFF;
  static Color get popupBackground => _colorFBFBFB;

  static Color get blackText => _color322D2D;
  static Color get darkText => _color000000_60;
}
```

```
3 String capitalize(String s) => s[0].toUpperCase() + s.substring(1);
4
5 String numberParsable(String s) => s.replaceAll(RegExp(','), '.');
6
```

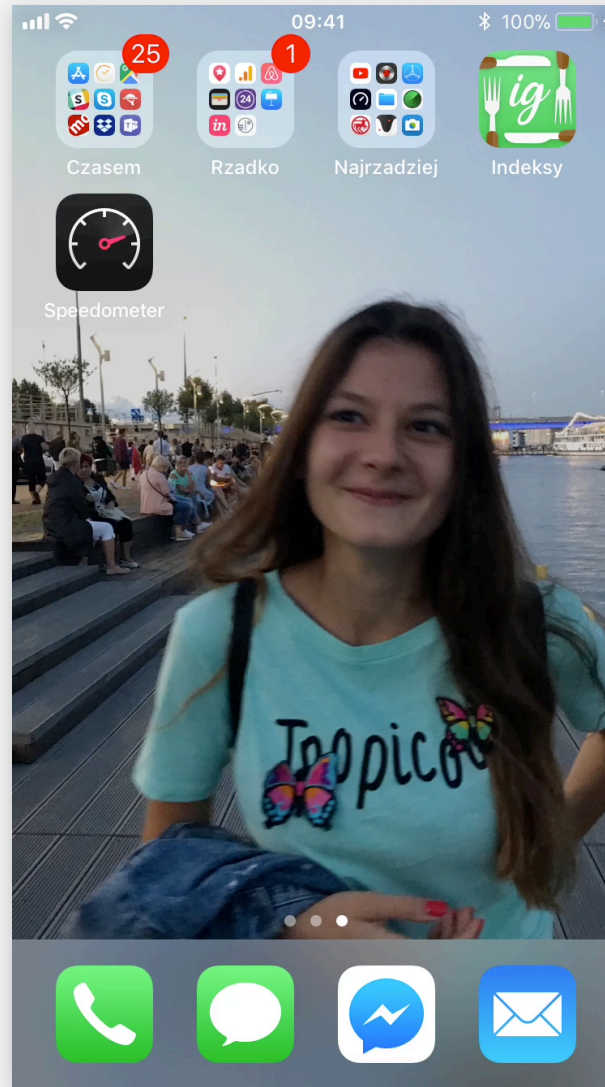
Issues

- TextField with centered cursor moves cursor back to center after typing a space.
- Double tap to select a word in text field does not work - [Issue #9288 \(Apr 2017\)](#)
 - work in progress - [PR #24034 \(iOS tap handling on CupertinoTextField\)](#)
- There is still a lot of work to be done in terms of components both for Android and iOS.

Android app launch on device

- _ Enable developer mode on the device. Settings > About > Tap 7 times on a build number which isn't named „Build Number” - different from device to device :) You'll see message you're a developer now.
- _ Go to developer settings and enable USB debugging, otherwise you'll not be able to install the app on the device and „flutter devices” will not show this device.
- _ At my first attempt to run the app on Android device I got a message that flutter is missing one dependency from default repositories (jcenter) - someone removed it and I had to change records in flutter SDK directory > packages > flutter_tools > gradle > flutter.gradle

Demo



Thanks!

Tomasz Szulc
Principal iOS Software Engineer

Blog: szulctomasz.com

Twitter: [tomkowz](https://twitter.com/tomkowz)

Instagram: [tomkowz_runner](https://www.instagram.com/tomkowz_runner)

